Exe 1, CRH to OWF. (10 points) Prove that the existence of collision-resistance hash function family (definition 12, lecture 5) implies the existence of one-way functions.

solution 1 As we did in HM2, exercise 3 (PRF imply PRG), lets make precise the notion of "randomly choose a function from the ensemble":
We have under discussion the ensemble $\{\mathcal{H}_n\}$, which we assume is a CRH. We know that it's an efficient function ensemble, so we know that there exist a PPT algorithm $D(1^n)$, that return a description of a function in $\{\mathcal{H}_n\}$. We assume that this description is a string $s \in \{0,1\}^*$. Together with D we have some other (deterministic) polynomial algorithm: $\mathsf{H}(1^n, s, x)$, that given s (the output of D), and some $x \in \{0,1\}^*$, returns $\mathsf{H}(1^n, s, x) \in \{0,1\}^n$. This is exactly the definition of a function in $\mathcal{H}_n$:

$$h_s : \{0,1\}^* \to \{0,1\}^n \qquad h_s(x) = \mathsf{H}(1^n, s, x)$$

Denote by p the polynomial that bounds the running time of D. Given a string $r \in \{0,1\}^{p(n)}$, we can assume that D is a deterministic algorithm acting as: $D(1^n, r)$, yielding a string s, which is a description of function in $\mathcal{H}_n$. So from now, by writing $h \leftarrow \mathcal{H}_n$, or saying "randomly choose a function from $\mathcal{H}_n$", we mean to chosen uniformly $r \leftarrow U_{p(n)}$ then compute: $s = D(1^n, r)$. We also denote that s, as s_r, emphasizing that this s, came from the randomness r.

Baring in mind those previous notations, and given a CRH ensemble $\{\mathcal{H}_n\}$, define the following function, that will be proved to be OWF:

$$f : \{0,1\}^{p(n)+2n} \to \{0,1\}^{p(n)+n} \qquad h_s(r \circ x) = h_{D(1^n,r)}(x) \circ D(1^n, r)$$

Some clarifications:

- The domain of f is actually a pair of strings. r of length $p(n)$, and an input x, from the domain of the functions of $\mathcal{H}_n$. Given an element in the domain of f, we always know how to break it to r and x, because we know their lengths.
- We use the notation $f(r, x)$ as $f(r \circ x)$
- The part r in the input string, serves as the randomness of $D(1^n, r)$. So randomly choose r from $\{0,1\}^{p(n)}$, is the same as randomly choose a function $h_s \leftarrow \mathcal{H}_n$. Thus the output of the function is $h_s(x) \circ s$. Since $|h_s(x)| = n$, by looking at an output of f we know what is the s (hence what is h_s)

Here is out main claim:

Claim 0.1. *The above f is OWF*

Proof of Claim 0.1. Assume not. So we have a PPT algorithm $A(1^{p(n)+2n}, y = f(r,x))$ a polynomial $q(n)$, and infinitely many n, such that:

$$\mathsf{Pr}_{r\leftarrow\{0,1\}^{p(n)},x\leftarrow\{0,1\}^{2n}}[(r',x') := A(1^{p(n)+2n}, f(r,x)) \quad : \quad (r',x') \in f^{-1}(f(r,x))] > \frac{1}{q(n)}$$

Lets simplify the above expression:

$$(r',x') \in f^{-1}(f(r,x)) \;\Leftrightarrow\; f(r',x') = f(r,x) \;\Leftrightarrow\; h_{D(1^n,r)}(x) \circ D(1^n,r) = h_{D(1^n,r')}(x') \circ D(1^n,r')$$

Denote $s_r = D(1^n, r)$, and $s_{r'} = D(1^n, r')$ we have:

$$h_{D(1^n,r)}(x) \circ D(1^n,r) = h_{D(1^n,r')}(x') \;\circ D(1^n,r') \;\Leftrightarrow h_{s_r}(x) = h_{s_{r'}}(x') \text{ and } s_r = s_{r'}$$

So we can write:

$$\mathsf{Pr}_{r\leftarrow\{0,1\}^{p(n)},x\leftarrow\{0,1\}^{2n}}[(r',x') := A(1^{p(n)+2n}, f(r,x)) \;:\; h_{s_r}(x) = h_{s_{r'}}(x') \text{ and } s_r = s_{r'}] > \frac{1}{q(n)}$$

We really don't care whether $s_r = s_{r'}$ or not. The real thing that is of interest for us is that $h_{s_r}(x) = h_{s_r}(x')$. So if A didn't succeed in inverting the pair (r,x), but somehow succeeded to produce a good x', it's OK with us. Since dropping the $s_r = s_{r'}$, just enlarges the event we are interested in, we get:

$$\mathsf{Pr}_{r\leftarrow\{0,1\}^{p(n)},x\leftarrow\{0,1\}^{2n}}[(r',x') := A(1^{p(n)+2n}, f(r,x)) \quad : \quad h_{s_r}(x) = h_{s_r}(x')] > \frac{1}{q(n)} \tag{1}$$

So lets have an intuitive discussion. (1) is the key point to our reduction. Given $h_s(x)$, if A is able to find an x' such that $h_s(x) = h_s(x')$, we come very close to break the hardness of $\mathcal{H}_n$. But still we have the problem that maybe $x = x'$. But since x, x' are chosen from the domain of $\{0,1\}^{2n}$, much larger than the range of f_s ($\{0,1\}^n$), we know that with a high probability: $x \neq x'$. That is enough to contradict the hardness of $\mathcal{H}_n$. Formally here is the proof for that:

Consider the following algorithm $B(1^n, s)$, which break the hardness of out PRH:

Algorithm 0.2 (B: break $\mathcal{H}_n$)**.**

input*: $1^n, s$ where f_s randomly chosen from $\mathcal{H}_n$*

- *uniformly sample: $x \leftarrow \{0,1\}^{2n}$*
- *let $r', x' := A(1^{p(n)+2n}, h_s(x) \circ s)$*
- *return $< x, x' >$*

So now, we need to show that the following probability is not negligible:

$$\mathsf{Pr}_{h_s \leftarrow \mathcal{H}_n}[(x, x') := B(1^n, s) \ : \ x \neq x' \text{ and } h_s(x) = h_s(x')]$$

By look at the definition of B we get that the above probability equals:

$$\mathsf{Pr}_{h_s \leftarrow \mathcal{H}_n, x \leftarrow \{0,1\}^{2n}}[(r', x') := A(1^{p(n)+2n}, h_s(x) \circ s) \ : \ x \neq x' \text{ and } h_s(x) = h_s(x')]$$

Writing it again, but instead of $h_s \leftarrow \mathcal{H}_n$, use $r \leftarrow \{0,1\}^{p(n)}$ we get:

$$\mathsf{Pr}_{r \leftarrow \{0,1\}^{p(n)}, x \leftarrow \{0,1\}^{2n}}[(r', x') := A(1^{p(n)+2n}, h_{s_r}(x) \circ s_r) \ : \ x \neq x' \text{ and } h_{s_r}(x) = h_{s_r}(x')]$$

Lets ease the notation by baring in mind the domain of the randomness, and the notation: $(r', x') := A(1^{p(n)+2n}, h_{s_r}(x) \circ s_r)$.
Denote the following events:

- E_1 is the event: $x \neq x'$
- E_2 is the event: $h_{s_r}(x) = h_{s_r}(x')$
- E_3 is the event: $h_{s_r}(x)$ has more that 1 source

We mainly interest in the situation that $h_{s_r}(x)$ has more that 1 source. Obviously we have:

$$\begin{aligned} \mathsf{Pr}[x \neq x' \text{ and } h_{s_r}(x) = h_{s_r}(x')] &= \mathsf{Pr}[E_1 \text{ and } E_2] \\ &\geq \mathsf{Pr}[E_1 \text{ and } E_2 \text{ and } E_3] \\ &= \mathsf{Pr}[E_1 \ \mid E_2 \text{ and } E_3] \ * \ \mathsf{Pr}[E_2 \text{ and } E_3] \end{aligned} \tag{2}$$

Now lets try to give lower bounds for the last 2 factors.

Claim 0.3. $\mathsf{Pr}[E_1 \ \mid E_2 \textit{ and } E_3] >= \frac{1}{2}$.

Proof of Claim 0.11. The following is obvious:

$$\mathsf{Pr}[x' = x \ \ |E_2 \text{ and } E_3] = \sum_{a \in \{0,1\}^{2n}} \mathsf{Pr}[x' = a \ \mid E_2 \text{ and } E_3 \text{ and } x = a] \cdot \mathsf{Pr}[x = a] \tag{3}$$

The elements in that sum for which values of a not fulfilling E_2 and E_3, are 0 so we ignore them. For the positive elements we notice that due to condition E_3, for each such a there exist at least one $a' \neq a$ such that $h_{s_r}(a) = h_{s_r}(a')$. Now consider algorithm A when it performs the following line:

let $r', x' := A(1^{p(n)+2n}, h_s(x) \circ s)$

For each such a, a', the call to $A(1^{p(n)+2n}, h_s(x) \circ s)$ is the same. Hence information theoretically we know that x' have the same distribution for both $x = a$ and $x = a'$. So the following equality holds:

$$\mathsf{Pr}[x' = a \ \ |E_2 \text{ and } E_3 \text{ and } x = a] \ = \ \mathsf{Pr}[x' = a' \ \ |E_2 \text{ and } E_3 \text{ and } x = a] \tag{4}$$

The following is also obvious:

$$\mathsf{Pr}[x' \neq x \ \ |E_2 \text{ and } E_3] \leq \sum_{a \in \{0,1\}^{2n}} \mathsf{Pr}[x' = a' \ \ | \ E_2 \text{ and } E_3 \text{ and } x = a] \cdot \mathsf{Pr}[x = a] \tag{5}$$

Note that it's only less or equal, because there may be values a, with a lot of corresponding a'. Using (4) we see that the sums in (3) and (5) are equal. Hence:

$$\mathsf{Pr}[x' \neq x \ \ |E_2 \text{ and } E_3] \geq \mathsf{Pr}[x' = x \ \ |E_2 \text{ and } E_3] \tag{6}$$

And the claim is proved. □

Claim 0.4. *There exist a polynomial $q_1(n)$ such that:* $\mathsf{Pr}[E_2 \textit{ and } E_3] \geq \frac{1}{q_1(n)}$.

Proof of Claim 0.4. Using the regular union bound technique we get:

$$\mathsf{Pr}[E_2 \text{ and } E_3] \geq \mathsf{Pr}[E_2] + \mathsf{Pr}[E_3] - 1 \tag{7}$$

Also it's quite easy to see that for every function $h : \{0,1\}^{2n} \to \{0,1\}^n$ we have:

$$\mathsf{Pr}_{x \leftarrow \{0,1\}^{2n}}[\left|h^{-1}(\{x\})\right| \geq 2] = 1 - neg(n)$$

Hence if follows that: $\mathsf{Pr}[E_3] = 1 - neg(n)$. The remaining work is to give a lower bound to the probability of E_2. But this is exactly what we had in (1). So we get that $\mathsf{Pr}[E_2] = \frac{1}{q(n)}$. Putting it into (7) we get:

$$\mathsf{Pr}[E_2 \text{ and } E_3] \geq \mathsf{Pr}[E_2] + \mathsf{Pr}[E_3] - 1 \ \geq \ \frac{1}{q(n)} + 1 - neg(n) - 1 \ \geq \ \frac{1}{q_1(n)}$$

For some polynomial $q_1(n)$ □

Putting the last 2 claims in (2), we get that:

$$\begin{aligned}\mathsf{Pr}[B \text{ succeed to break } \mathcal{H}_n] &= \mathsf{Pr}[x \neq x' \text{ and } h_{s_r}(x) = h_{s_r}(x')] \\ &\geq \mathsf{Pr}[E_1 \ \ | \ E_2 \text{ and } E_3] \ * \ \mathsf{Pr}[E_2 \text{ and } E_3] \\ &\geq \frac{1}{2} \cdot \frac{1}{q_1(n)}\end{aligned}$$

Contradicting the hardness for $\mathcal{H}_n$ □

Exe 2, Birthday paradox (10 points). Prove that $\mathsf{Pr}_{\pi\leftarrow\Pi_n}\left[\exists x \neq x' \in \mathcal{S} \colon \pi(x) = \pi(x')\right] \in \Omega(1)$, where $\mathcal{S} \subset \{0,1\}^n$ is of size $2^{n/2}$ (n is a power two).

You might find the following inequality useful: $e^{-x} \geq (1-x)$ for $x \in [0,1]$

solution 2 Given $\mathcal{S} \subset \{0,1\}^n$ of size $2^{n/2}$, denote it by: $\mathcal{S} = \{s_1, s_2, \ldots, s_{2^{\frac{n}{2}}}\}$.
For $\pi \in \Pi_n$, define the events A_i and B_i for $1 \leq i \leq 2^{n/2}$:
A_i: $\{\pi(s_1), \ldots, \pi(s_i)\}$ - are different
B_i: $\pi(s_i) \notin \{\pi(s_1), \ldots, \pi(s_{i-1})\}$

We have:

$$\begin{aligned}
\mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_i] &= \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_i|\ A_{i-1}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_{i-1}] \\
&= \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_i|\ A_{i-1}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_{i-1} \mid A_{i-2}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_{i-2}] \\
&= \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_i|\ A_{i-1}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_{i-1} \mid A_{i-2}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_{i-2} \mid A_{i-3}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_{i-3}] \\
&= \ldots \\
&= \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_i|\ A_{i-1}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_{i-1} \mid A_{i-2}] \cdot \ldots \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_2 \mid A_1]
\end{aligned}$$

Since it's obvious that:

$$\mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_j|\ A_{j-1}] = \frac{2^n - (j-1)}{2^n}$$

We get:

$$\begin{aligned}
\mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_i] &= \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_i|\ A_{i-1}] \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_{i-1} \mid A_{i-2}] \cdot \ldots \cdot \mathsf{Pr}_{\pi\leftarrow\Pi_n}[B_2 \mid A_1] \\
&= \frac{2^n - (i-1)}{2^n} \cdot \frac{2^n - (i-2)}{2^n} \cdot \ldots \cdot \frac{2^n - 1}{2^n} \\
&= (1 - \frac{i-1}{2^n}) \cdot (1 - \frac{i-2}{2^n}) \cdot \ldots \cdot (1 - \frac{1}{2^n}) \\
&\leq e^{-\frac{i-1}{2^n}} \cdot e^{-\frac{i-2}{2^n}} \cdot \ldots \cdot e^{-\frac{1}{2^n}} \\
&= e^{-(\frac{i-1}{2^n} + \frac{i-2}{2^n} + \ldots + \frac{1}{2^n})} = e^{-\frac{i\cdot(i-1)}{2\cdot 2^n}}
\end{aligned}$$

Where we used the well known fact: $e^{-x} \geq (1-x)$. Putting $i = 2^{n/2}$, we get that:

$$\mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_{2^{\frac{n}{2}}}] = e^{-\frac{2^{n/2}\cdot(2^{n/2}-1)}{2\cdot 2^n}} = e^{-\frac{2^n - 2^{n/2}}{2\cdot 2^n}} = e^{-\frac{1}{2} + \frac{2^{n/2}}{2\cdot 2^n}}$$

Hence for $n \geq 4$ we have:

$$\mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_{2^{\frac{n}{2}}}] = e^{-\frac{1}{2} + \frac{2^{n/2}}{2\cdot 2^n}} \leq e^{-\frac{1}{2} + \frac{4}{2\cdot 16}} = e^{-\frac{1}{2} + \frac{1}{8}} = e^{-\frac{3}{8}}$$

Back to what we need to prove:

$$\begin{aligned}
\mathsf{Pr}_{\pi\leftarrow\Pi_n}\left[\exists x \neq x' \in \mathcal{S} \colon \pi(x) = \pi(x')\right] &= 1 - \mathsf{Pr}_{\pi\leftarrow\Pi_n}[\forall x \neq x' \in \mathcal{S} \colon \pi(x) \neq \pi(x')] \\
&= 1 - \mathsf{Pr}_{\pi\leftarrow\Pi_n}[A_{2^{\frac{n}{2}}}] \\
&\geq 1 - e^{-\frac{3}{8}} \in \Omega(1)
\end{aligned}$$

Exe 3, Interactive Proofs, Goldreich, Chapter 5, exe 2, (10 points) Prove that if $\mathcal{L}$ has an interactive proof system with *deterministic* verifier, then $\mathcal{L} \in$ NP. Guideline: note that if the verifier is deterministic, then the entire interaction between the prover and verifier can be determined by the prover.

Exe 3 Assume (P, V) is an interactive proof for $\mathcal{L}$, where V is a deterministic. The first step is the following claim

Claim 0.5. *There exist a deterministic prover P^{det}, such that (p^{det}, V) (same deterministic V as given) is interactive proof for $\mathcal{L}$*

Proof of Claim 0.5. The idea behind removing P randomness is to use the property that the prover can be all powerful machine, so it can simulate P, V and all the possibilities of P's randomness
So suppose $x \in \mathcal{L}$. If $r \in \{0,1\}^l$ is a randomness of P, P^{det} can simulate $P(r), V$ when interaction on x deterministically. At the end of a specific simulation either:

1. P^{det} exhaust r - so it quits this specific simulation, and move to the next one (with different r ...)
2. V rejected the input. In this case also P^{det} move to the next simulation
3. V accepts. So P^{det} writes all the interaction he made with V during this simulation, and use it in the 'real' simulation with V

It's obvious that P^{det} can scan all possible randomness strings in $\{0,1\}^*$, simply in increasing order of length, and for each length l scan all 2^l possibilities. Since (P, V) accepts x with positive probability (even higher than 2/3), we know that P^{det} will find a good random sequence, and will be able to use it when it interacts with V. (this interaction will make V to accept because V is deterministic) $\square$

So from now we'll assume P is deterministic.

Claim 0.6. (P, V) *have perfect completeness. that is:*

$$\forall x \in \mathcal{L} \ \ \mathsf{Pr}[< (\mathrm{P}, \mathrm{V})(x) >= 1] = 1$$

Proof of Claim 0.6. Since both P and V are deterministic, we get that for every $x \in \mathcal{L}$ either

1. $\mathsf{Pr}[< (\mathrm{P}, \mathrm{V})(x) >= 1] = 1$
2. $\mathsf{Pr}[< (\mathrm{P}, \mathrm{V})(x) >= 1] = 0$

Obviously the second option is impossible since $\mathsf{Pr}[< (\mathrm{P}, \mathrm{V})(x) >= 1] \geq 2/3$ $\square$

Claim 0.7. (P, V) *have perfect soundness. that is:*

$$\textit{For any algorithm } P^* \textit{ and } \forall x \notin \mathcal{L} \ \ \mathsf{Pr}[< (\mathrm{P}^*, \mathrm{V})(x) >= 1] = 0$$

Proof of Claim 0.7. Assume on the contrary that there exist an algorithm P^*, and input $x \notin \mathcal{L}$ such that:

$$\mathsf{Pr}[<(\mathrm{P}^*, \mathrm{V})(x)> = 1] \ > \ 0$$

The same way as we proved Claim 0.5, we can prove that there exist a deterministic cheater P^*_{det}, that interacting with V on x will make V accept x. So for that P^*_{det}, and that specific x we get

$$\mathsf{Pr}[<(\mathrm{P}^*_{det}, \mathrm{V})(x)> = 1] \ = \ 1 > \frac{1}{3}$$

Contradiction. □

Now we can prove out main claim:

Claim 0.8. $\mathcal{L} \in NP$

Proof of Claim 0.8. Consider the (deterministic) interactive proof (P, V) we have for $\mathcal{L}$. Assume this is a k-steps interaction proof, so for every $x \in \mathcal{L}$ we have the following $2k$ messages (actually strings) passes between P and V: $p_1(x), v_1(x), \ldots, p_k(x), v_k(x)$.

We define the witness of $x \in \mathcal{L}$ as:

$$w(x) \ = \ p_1(x)\#p_2(x)\#\ldots\#p_k(x) \quad \text{(As usual we assume '\#' is a new symbol).}$$

We define a deterministic polynomial algorithm $A(x, w(x))$ that fulfil:

1. $\forall x \in \mathcal{L} \quad A(x, w(x)) = 1$
2. $\forall x \notin \mathcal{L}$ and $\forall w' \in \{0,1\}^* \quad A(x, w') = 0$

The algorithm:

Algorithm 0.9 $(A(x, w(x)))$**.**

input: x , $w(x) = p_1(x)\#p_2(x)\#\ldots\#p_k(x)$

- *Start an interaction with V, on input x.*
- *for ($i = 1$ to k) do:*
- *Send V the message $p_i(x)$*
- *wait for V to response*
- *if (V accepts)*
- *return 1*
- *else*
- *return 0*

Due to Claim 0.6 it's obvious that $\forall x \in \mathcal{L} \quad A(x, w(x)) = 1$. The existence of an input $x \notin \mathcal{L}$, and a fake witness w', such that $A(x, w') = 1$, will contradict Claim 0.7. □

Exe 4, Zero knowledge (10 points) Prove that the interactive proof presented in class for graph non-isomorphism is *honest-verifier* perfect zero-knowledge (i.e., the ZK definition is restricted to $\mathsf{V}^* = \mathsf{V}$).

Bonus (5 points): Is the above protocol (full fledged) zero knowledge? justify your answer as good as you can.

solution 4 Denote by (P, V) the interactive prove we saw in class for GNI. Here is a PPT algorithm S, simulate perfectly the interaction (P, V) on an input $x \in GNI$

Algorithm 0.10 (S simulator for (P, V)).

<u>input</u>: $x = (G_0 = ([m], E_0)\ ,\ G_1 = ([m], E_1))$ *where* G_1 *not isomorphic to* G_2

- *choose* $b \leftarrow \{0, 1\}$ *and* $\pi \leftarrow \Pi_m$
- *set* $\pi(E_b)$ *the value sent to* P
- *Set* $b' = b$
- *exit*

. . . .

Claim 0.11. *For every non isomorphic pair of graphs* $x = (G_0 = ([m], E_0)\ ,\ G_1 = ([m], E_1))$, *we have:*

$$\{< (P, V)(x) >\} \approx_{Perfect} \{S(x)\}$$

Proof of Claim 0.11. Since b and π are defined the same in S and V, it's obvious that they and the graph sent to P have the same distributions in S and V. Since we assume that the input $x \in NIG$, we know that b' returned by P will be equal to b, chosen by V. Hence also b' of V and b' of S have the same distribution. To summarize, we get that all the variables distributed the same under $(P, V)(x)$ and under $S(x)$, hence the claim follows ☐

Bonus The IP isn't ZK. Before we explain it, let's assume the following:

- In case that the verifier sends to the prover a graph that isn't isomorphic to both G_1,G_2, the prover quits. (Hence the verifier knows the fact that the graph he sent isn't isomorphic to G_1,G_2)
- The (cheating) verifier V^*, can get an input given to it before it starts to communicate with P. So assume it get a graph G', denote this fact as $V^*_{G'}$. In this case, a simulator for $V^*_{G'}$, will also get the same input graph: $S_{G'}$.

Consider the following cheating verifier:

Algorithm 0.12 (V_G^*, where $G =< V, E >$)**.**

input: $x = (G_0 = ([m], E_0)\ ,\ G_1 = ([m], E_1))$

- *send* E *to* P
- *if (*P *aborts)*
- *set isomorphic = false*
- *else*
- *set isomorphic = true*

Assuming that the above protocol is ZK, we conclude that there is a simulator S_G that simulate V_G^* for every $x = (G_0, G_1)$ non isomorphic graphs:

$$\text{view of } S_G(x) \ \approx_c \ \text{view of } V_G^*(x)$$

Using this simulator we can get a BPP algorithm for solving the GI problem:

Algorithm 0.13 (B: Solve GI problem)**.**

input: $x = (G' = (V, E')\ ,\ G'' = (V, E''))$

- *define* $\widetilde{G'}$ *to be a graph not isomorphic to* G' *(remove or add an edge)*
- *call to* $S_{G'}(\widetilde{G'}, G'')$
- *if(* $S.isomorphic = false$*)*
- *decide* G' *not isomorphic to* G''
- *else*
- *decide* G' *isomorphic to* G''

Where by $S.isomorphics$ we mean the variable *isomorphic* of V^*. Notice that since $\widetilde{G'}$ isn't isomorphic to G', the prover quits iff G' isn't isomorphic to G''. Hence out algorithm B will return with probability (close to) 2/3, the correct answer.